AMENDMENTS TO THE CLAIMS

Listing of Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method comprising:

generating a record of software system events, each event record within the record of system events representing an inter-component control or dataflow interaction;

creating a behavioral template based on a predetermined behavior of the software system, wherein the predetermined behavior comprises a predetermined set of state changes selected from an execution of the software system, wherein the predetermined set of state changes represent coherent units of behavior by the software system;

identifying an occurrence of the predetermined behavior within the record of software system events, based on the behavioral template and wherein the predetermined set of state changes may be used as a behavioral model for a debugger to recognize;

determining that the behavioral model is too specific; and

replacing a found instance of the predetermined behavior with a replacement sequence of events, wherein the replacement sequence of events is an abstract event of a higher level than one or more system events that comprise the predetermined behavior.

2. (Original) A method according to claim 1 wherein creating the behavioral template comprises creating a visual prototype, which represents the predetermined behavior of the

software system.

3. (Original) A method according to claim 1 wherein creating the behavioral template comprises creating a behavior expression, which represents the predetermined behavior of the software system.

4. (Original) A method according to claim 1 wherein generating the record of software system events comprises simulating an execution of the software system, with the record of software system events generated by the simulator.

5. (Previously Presented) A method according to claim 1 wherein generating the record of software system events comprises:

instrumenting the software system to provide an event notification to a runtime operating system for each software system event;

deploying the software system to a target architecture; and

on the target architecture, capturing all notifications from the software system and storing the event notifications to create a record of software system events.

Claims 6-17. (Canceled).

18. (Currently Amended) A software system design tool comprising:

a simulator for simulating an execution of the software system;

a template tool for creating a behavioral template based on a predetermined behavior of the software system, wherein the predetermined behavior comprises a predetermined set of state changes selected from an execution of the software system, wherein the predetermined set of state changes represent coherent units of behavior by the software system and wherein the predetermined set of state changes may be used as a behavioral model for a debugger to recognize and wherein the behavioral model is determined to be too specific; and

a debugging tool for identifying an instance of the predetermined behavior of the software system from a simulated execution of the software system based on the behavioral template,

wherein a found instance of the predetermined behavior is replaced with a replacement sequence of events, and wherein the replacement sequence of events is an abstract event of a higher level than one or more system events that comprise the predetermined behavior.

19. (Original) A software system design tool according to claim 18 wherein the template tool allows a designer to create a behavioral template based on a visual prototype.

20. (Original) A software system design tool according to claim 18 wherein the template tool allows a designer to create a behavioral template based on a behavior expression.

21-30. (Canceled).

31. (Currently Amended) A method comprising:

generating a record of software system events, each event record within the record of system events representing an inter-component control or dataflow interaction;

creating a behavioral template based on a predetermined behavior of the software system, wherein the predetermined behavior comprises a predetermined set of message events selected from an execution of the software system, wherein the predetermined set of message events represent coherent units of behavior by the software system and wherein a predetermined set of state changes may be used as a behavioral model for a debugger to recognize;

determining that the behavioral model is too specific;

identifying an occurrence of the predetermined behavior within the record of software system events, based on the behavioral template; and

replacing a found instance of the predetermined behavior with a replacement sequence of events, wherein the replacement sequence of events is an abstract event of a higher level than the system events that comprise the predetermined behavior.

32. (Previously Presented) A method according to claim 31 wherein creating the behavioral template comprises creating a visual prototype, which represents the predetermined behavior of the software system.

33. (Previously Presented) A method according to claim 31 wherein creating the behavioral template comprises creating a behavior expression, which represents the predetermined behavior of the software system.

34. (Previously Presented) A method according to claim 31 wherein generating the record of software system events comprises simulating an execution of the software system, with the record of software system events generated by the simulator.

35. (Previously Presented) A method according to claim 31 wherein generating the record of software system events comprises:

instrumenting the software system to provide an event notification to a runtime operating system for each software system event;

deploying the software system to a target architecture; and

on the target architecture, capturing all notifications from the software system and storing the event notifications to create a record of software system events.

36. (Currently Amended) A software system design tool comprising:

a simulator for simulating an execution of the software system;

a template tool for creating a behavioral template based on a predetermined behavior of the software system, wherein the predetermined behavior comprises a predetermined set of message events selected from an execution of the software system, wherein the predetermined set of message events represent coherent units of behavior by the software system and wherein a predetermined set of state changes may be used as a behavioral model for a debugger to recognize and wherein the behavioral model is determined to be too specific; and

a debugging tool for identifying an instance of the predetermined behavior of the software system from a simulated execution of the software system based on the behavioral template,

wherein a found instance of the predetermined behavior is replaced with a replacement sequence of events, and wherein the replacement sequence of events is an abstract event of a higher level than one or more system events that comprise the predetermined behavior.

37. (Previously Presented) A software system design tool according to claim 36 wherein the template tool allows a designer to create a behavioral template based on a visual prototype.

38. (Previously Presented) A software system design tool according to claim 36 wherein the template tool allows a designer to create a behavioral template based on a behavior expression.